

Detection of Web Propaganda Patterns by Transformer Neural Networks: Improving Efficiency via Dataset Balancing

Maryna Molchanova^{1*}, Volodymyr Didur¹, Olena Sobko¹ and Olexander Mazurets¹

¹ Khmelnytskyi National University, 11, Instytut's'ka str., Khmelnytskyi, 29016, Ukraine

Abstract

In the paper, a proposed approach for improving efficiency of web propaganda patterns detection by transformer neural networks is presented. Approach consists of sequential use of three developed methods: method for dataset balancing, method for fine-tuning individual binary neural network models and method for detecting web propaganda patterns. Compared to existing analogues, the use of proposed approach allowed achieving an efficiency increase of 0.1 by F1 metric when detecting propaganda patterns in web texts using transformer neural networks due to dataset balancing optimization. Analyzing the impact of parameter that determines proportion of texts without web propaganda patterns allows assessing how the models ability to distinguish propaganda patterns from neutral texts and texts with other propaganda patterns. This allows finding the optimal ratio of dataset classes to increase the overall effectiveness for detecting web propaganda patterns. Conducted research has established that the highest results were achieved when forming the training dataset with a percentage of texts without patterns of 30% using the RoBERTa neural network, and was achieved 0.725 by F1 metric. Proposed approach ensures the determination of the optimal ratio between text sets with propaganda patterns and neutral text set, which improving the generalization ability of models and reduce their bias.

Keywords

web propaganda patterns, dataset balancing, BERT, RoBERTa, NLP, transformer neural network

1. Introduction

In the modern information environment, propaganda content plays a significant role in shaping public opinion, political views, and social behavior [1]. Social networks have become a key space for disseminating information, but at the same time they are also a tool for manipulative influence [2]. Algorithmic content distribution, personalized news feeds, and automated recommendation systems contribute to the rapid spread of manipulative messages, which makes it difficult to detect web propaganda patterns using traditional methods [3, 4]. Since manipulative content can have subtle linguistic markers and adapt to the context [5], its identification requires the use of context-oriented language models, in particular transformers [6]. Significant progress in the field of automatic text analysis has made it possible to use neural networks to detect manipulations, but the accuracy of such models largely depends on the training sample. The balance of the sample affects the model's ability to recognize manipulative patterns and distinguish them from neutral or unintentional influence [7].

The research is closely related to the UN Sustainable Development Goals, as it contributes to the formation of quality education (SDG No. 4) through the development of media literacy and critical thinking [8]. This allows society to more effectively recognize manipulative content and make informed decisions, which is consistent with the principles of ensuring access to reliable information. In addition, methods for detecting web propaganda patterns in text messages play an important role in maintaining peace, justice and strengthening democratic institutions (SDG No. 16) [9, 10]. They help combat disinformation, increase the level of transparency of governance and contribute to reducing the impact of manipulation in society, which is a key factor in the sustainable development of the information space [11].

¹CMIS-2025: Eighth International Workshop on Computer Modeling and Intelligent Systems, May 5, 2025, Zaporizhzhia, Ukraine

✉ m.o.molchanova@gmail.com (M. Molchanova); pravetz@ukr.net (V. Didur); olenasobko.ua@gmail.com (O. Sobko); exe.chong@gmail.com (O. Mazurets)



0000-0001-9810-936X (M. Molchanova); 0009-0008-2279-1487 (V. Didur); 0000-0001-5371-5788 (O. Sobko); 0000-0002-8900-0650 (O. Mazurets)



© 2025 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The aim of paper is to improve the efficiency of detecting propaganda patterns in web texts using transformative neural networks by optimizing the dataset balancing. Research is aimed at reducing the impact of class imbalance, increasing the accuracy of classification and improving the generalization ability of model.

The main paper contribution is created methodology that includes method for fine-tuning individual binary neural network models to detect propaganda patterns, method for balancing the dataset, and method for detecting web propaganda patterns. The paper also provides an analysis of the impact of the balance of the training sample on the effectiveness of models for detecting manipulative patterns in social media. An experimental study of the performance of the BERT and RoBERTa transformative language models depending on the distribution of training examples between classes was conducted. The results obtained contribute to a deeper understanding of the role of the training sample in improving algorithms for detecting manipulative texts and can be used to increase the reliability of automated systems for analyzing the information space.

2. Related Works

The issue of automated detection of web propaganda patterns in social media has widely attracted the attention of researchers.

The research [12] considers a multimodal and multilingual dataset of propaganda patterns PPN (Propagandist Pseudo-News), which contains news texts collected from web resources that expert organizations have classified as containing manipulation patterns. The study analyzes various NLP approaches that allow identifying the characteristic features that annotators have highlighted and comparing them with the results of automated classification. For this purpose, the following methods are used: VAGO to determine the level of subjectivity and vagueness of statements, TF-IDF as a basic analysis tool, as well as four classification algorithms – two RoBERTa models, CATS, which focuses on syntactic features, and XGBoost, which combines semantic and syntactic features.

In [13] two architectures for classifying propaganda patterns were analyzed: one involved the use of data augmentation (EDA) methods, and the other worked without them. The models using EDA showed a 3% improvement in F_1 -measure, reaching 57.57% on the test set. A significant increase in accuracy was observed for manipulation patterns such as "Appeal_to_fear-prejudice", "Exaggeration, Minimisation" and "Repetition", while for individual techniques, in particular "Doubt" and "Flag-Waving", a slight decrease in results was noted. "Causal_Oversimplification" and "Thought-terminating_Cliches" showed the most noticeable improvement. Determination of optimal parameters for classification was carried out by analyzing the number of epochs, the length of text fragments and the learning rate. This allowed the authors to achieve an F_1 -measure of 44% in the sentiment detection task and 57% in the classification of manipulation patterns.

The authors of [14] used the RoBERTa language model to detect propaganda patterns in news articles. The model was evaluated on the SemEval-2020 Task 11 reference dataset, which confirmed its effectiveness in recognizing complex manipulation patterns in text. Compared to baseline model, RoBERTa achieved an F_1 -measure of 60.2%, demonstrating its higher accuracy.

In [15] the multilingual set of propaganda patterns was created by translating the PTC and WANLP corpora, supplemented with SemEval23 data. Three models were proposed: MultiProp-Baseline (an ensemble of GPT-2, mBART and XLM-RoBERTa), MultiProp-ML (meta-learning for languages with minimal data) and MultiProp-Chunk (processing long texts exceeding the token limit). As a result of the experiments, the F_1 score for the Polish language was 62.5%.

The study [16] indicates the ambiguity in the ability of LLMs to recognize propaganda patterns in news texts. Experiments conducted on the annotated SemEval2020 Task 11 corpora demonstrated maximum Recall values of 64.53% and Precision of 81.82%. At the same time, none of the models was able to exceed the baseline F_1 score, which was approximately 50%. The highest achieved F_1 score was only 20%, which is significantly inferior to the baseline and indicates the limitations of generative models in ensuring reproducibility.

In [17] emphasize that most previous studies focused on linguistic features to detect manipulation patterns in texts. Therefore, authors propose the method based on meta-learning that allows for automatic identification of semantic manipulation patterns at sentence level in news materials. For this, multi-task learning is used, aimed at detecting semantic contradictions. Proposed approach combines CRF, BiLSTM and pre-trained language models, which provides an F_1 -measure of 61% for multilingual data and 68.8% for monolingual.

The authors of [18] evaluate the possibility of using large language models (LLMs), in particular OpenAI GPT-3.5-turbo, to detect propaganda in news articles. The analysis is based on 18 propaganda techniques identified by Martino et al., and covers materials from Russia Today and the SemEval-2020 Task 11 corpus. Using a specially designed prompt, the model determines the presence of propaganda techniques and classifies articles. Qualitative analysis of results allows us to assess effectiveness of LLMs in this task and optimal prompt parameters.

The application of machine learning models to identify manipulation patterns in text content is considered in the study [19]. Among the analyzed approaches, the Stacking Classifier, which uses feature processing methods, in particular Word2Vec and TF-IDF, demonstrates high adaptability and accuracy. Comparative analysis shows that this model outperforms others, such as Naive Bayes, SVM, KNN, Logistic Regression and Random Forest. The implementation of feature engineering significantly improves the results, which is confirmed by the increase in Accuracy, Precision and F₁-measure.

The study [20] considers the application of machine learning methods to detect types of propaganda in the text content of social networks. The authors used data obtained through the social network API to evaluate the effectiveness of various models. The results of the study showed that neural networks, in particular the LSTM architecture, have high accuracy in this task, reaching 77.15%. It is noted that the further implementation of more modern models, such as BERT, can contribute to even better results in future studies.

Paper [21] proposes an ensemble model for identifying manipulation patterns in texts obtained from memes. The authors consider the use of modern pre-trained language models, as well as optimization methods, in particular data augmentation and combining multiple models. The model evaluation was carried out on the SemEval-2021 Task 6 dataset, and the results showed that proposed approach allows achieving an F₁-micro measure of 60.4% on the test set.

Authors of [22] used a two-stage process to determine the optimal threshold for classifying manipulation patterns to assess the effectiveness of the model. First, experiments were conducted with macrothresholds in the range from 0.1 to 0.9, the threshold with the highest F₁ score was selected, after which microthresholds were added for further optimization. The XLM-RoBERTa models were trained using the Adam optimizer, and early termination was used to prevent overtraining. The Accuracy, Precision, Recall, and F₁-measure metrics were used to assess performance at each stage.

From above reviews of scientific publications, it is clear that the issue of balancing datasets in existing methodologies was considered only from the perspective of creating synthetic samples, and the issue of the influence of the number of texts without manifestations of propaganda patterns was not considered at all. Therefore, our study is relevant and aims to eliminate this drawback by analyzing the influence of the number of texts without propaganda patterns on the effectiveness of transformer models.

The paper aims to determine the optimal ratio between texts with propaganda patterns and neutral texts, which will improve the generalizability of the models and reduce their bias.

3. Methodology

To solve the problem of detecting web propaganda patterns, it is first necessary to fine-tune the neural networks to detect each of the web propaganda patterns. Accordingly, this can be formalized as the problem of training a set of individual binary neural network models NN , where each model nn_i corresponds to a certain propaganda pattern p_i from the set of propaganda patterns P :

$$P = \{p_1, p_2, \dots, p_k\}, \quad (1)$$

where p_i – i -th propaganda pattern, k – number of unique propaganda patterns, $i=1..k$. Within the scope of the study, $k=10$, and the set P acquires the following elements:

- p_1 = "Loaded Language";
- p_2 = "Glittering Generalities";
- p_3 = "Euphoria";
- p_4 = "Appeal to Fear";
- p_5 = "FUD";
- p_6 = "Bandwagon";
- p_7 = "Thought-Terminating Cliche";

- p_8 ="Whataboutism";
- p_9 ="Cherry Picking";
- p_{10} ="Straw Man".

This set of propaganda patterns is linked to the existing data source presented within the framework of UNLP 2025 [23], dedicated to the competition for detecting manipulative propaganda patterns in the Ukrainian-language media space [24].

Accordingly, $\{NN\}$ will take the form:

$$NN = \{nn_1, nn_2, \dots, nn_k\}, \quad (2)$$

where nn_i – i -th neural network for i -th propaganda pattern.

Approach for detection of web propaganda patterns by transformer neural networks consists of sequential use of three developed methods: method for dataset balancing, method for fine-tuning individual binary neural network models and method for detecting web propaganda patterns (Figure 1).

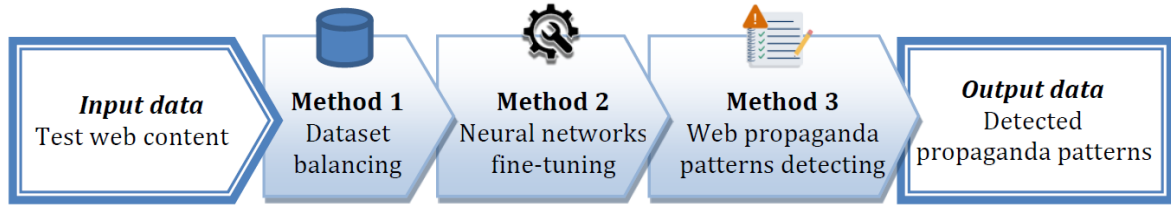


Figure 1: Sequence of methods execution in approach for web propaganda patterns detection

Proposed approach ensures the determination of the optimal ratio between text sets with propaganda patterns and neutral text set, which improving the generalization ability of models and reduce their bias. This improves the efficiency of detecting propaganda patterns in web texts using transformer neural networks through optimizing the dataset balancing.

3.1. Method for Dataset Balancing

Method for dataset balancing is designed to transform the general set of data in the input dataset into 2 datasets (training dataset and validation dataset), which will allow to increase the accuracy of detecting propaganda patterns in web texts. Scheme of training dataset prepare is shown in Figure 2.

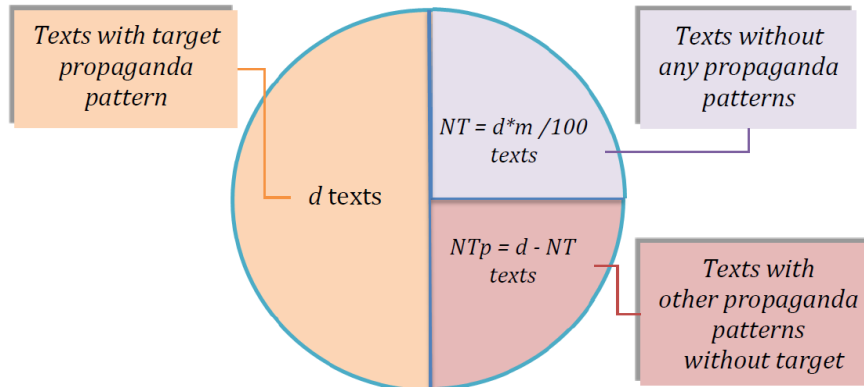


Figure 2: Scheme of training dataset prepare

Percent of texts without manipulation patterns m – the studied parameter for analyzing the influence of the balance of the training sample on the effectiveness of models for detecting manipulative propaganda patterns in social media. This parameter has an impact on the formation of the training dataset.

In addition to the training dataset, a validation dataset is constructed, which consists equally of all types of web propaganda patterns and texts without propaganda. This allows determining whether the model does not confuse patterns with each other and whether it is able to detect them independently of each other, which is critically important for the multi-label classification problem.

Accordingly, the result of the method of dataset balancing will be 2 datasets: training dataset and validation dataset. Schematically, their composition is shown in Figure 3.

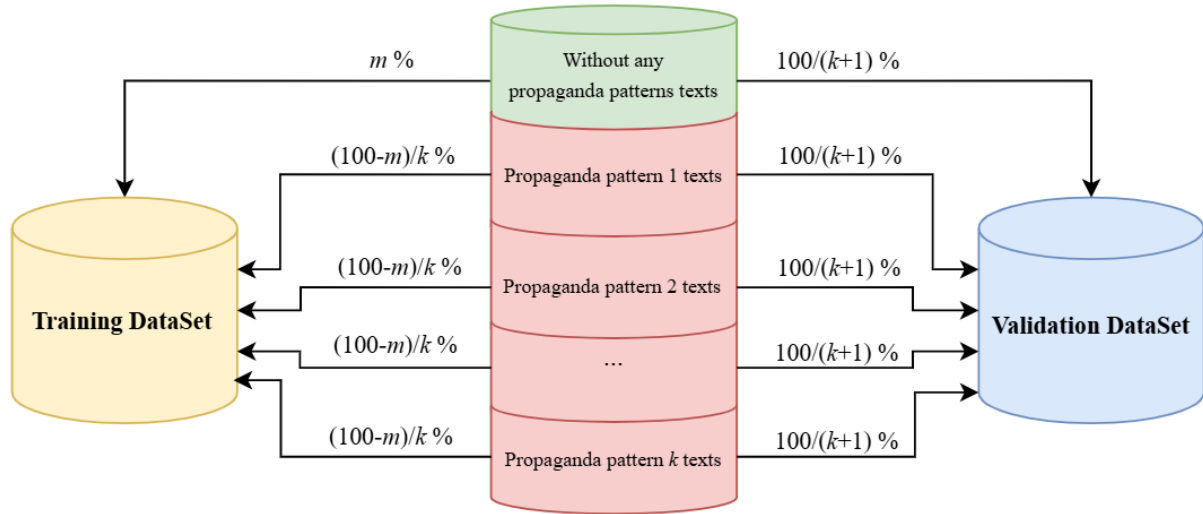


Figure 3: Scheme of datasets balancing for improve accuracy of pattern classification

It is worth noting that the base dataset is annotated at the fragment level, and the training dataset and validation dataset contain not the full text, but fragments (sentences that are marked as propaganda patterns).

The dataset contains annotated data at the fragment level that determine the presence of manipulative influence patterns from the set P . A typical text of the dataset from the category "propaganda patterns" can have either one or several labels. A typical text of the dataset from the category "without propaganda patterns" does not contain any web propaganda patterns from the set P . According to the marked data, the number of documents corresponding to the patterns $p_1 - p_{10}$ has the distribution shown in Table 1.

Table 1
Distribution of labeled data in dataset by categories (with and without manipulation patterns)

Propaganda patterns	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	without manipulation patterns
Document count	1973	483	462	300	385	157	463	158	512	138	1233

This approach to dataset generation allows us to assess the impact of sample balancing on the quality of propaganda pattern detection, as well as to avoid the dominance of the most common classes in the training set [25]. Using separate binary models for each pattern allows us to model them independently, which is important in problems with class intersection, when one text may contain several types of manipulation. This allows us to investigate how each pattern is separated within the data corpus and how it is affected by the imbalance of the training sample.

3.2. Method for Fine-Tuning Individual Binary Neural Network Model for Propaganda Patterns Detection

As can be seen from Table 1, the data have an uneven distribution, so using a single multi-class neural network model will not allow to obtain high results. A multi-class model tends to dominate widely represented classes, which leads to a decrease in accuracy for poorly represented classes. As a result, the model may simply ignore small categories, which will lead to a significant imbalance in predictions. In addition, multi-class classification assumes that the text belongs to only one class [26], which contradicts the nature of the task, where 1 text can have several labels corresponding to certain web propaganda patterns. Accordingly, using separate binary models for each p_i pattern allows to train each model separately without the influence of the imbalance of other classes to take

into account texts with several patterns, since each model from NN set works independently and does not limit the choice to only one class.

To investigate the impact of the balance of the training sample on the detection of web propaganda patterns using a set of individual binary neural network models NN , it is necessary to first present a method for obtaining a typical individual binary neural network model nn_i for detecting propaganda pattern p_i , the scheme of which is shown in Figure 4.

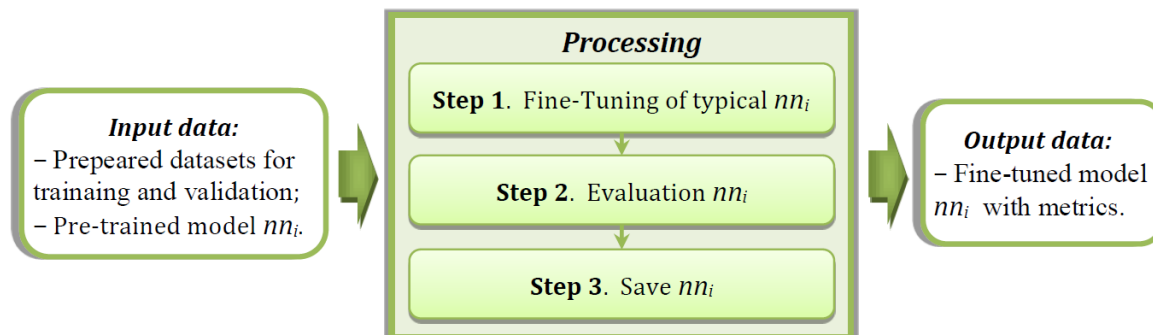


Figure 4: Scheme of method for fine-tuning individual binary neural network model

The input data of the method are prepared datasets for training and validation and pre-trained model nn . On Step 1, Fine-Tuning of typical nn_i on training DataSet, formed by method of datasets balancing, is performed. Fine-Tuning within the framework of the study will be carried out for individual binary neural network model BERT [27] and RoBERTa [28] with «HuggingFace» library [29].

Accordingly, on Step 2, evaluation of individual binary neural network model nn_i is performed, for evaluations both training dataset and validation dataset, which were formed by method of datasets balancing, will be used. Evaluation of models will be carried out by metrics Accuracy, Precision, Recall and F_1 . On Step 3, save of validated nn_i is performed. Accordingly, output data is fine-tuned model nn_i with metrics.

As pre-trained model nn , the use of BERT-like architectures is proposed, since these models can be applied to the analysis of Ukrainian texts even in the absence of large volumes of marked-up data [30, 31]. This feature is associated with pre-training on large text corpora, which allows these models to form universal language representations that can be refined on specific datasets to detect propaganda patterns. Fine-tuning allows you to adapt the model to the specifics of manipulative discourse, in particular in the Ukrainian language environment, which contains both unique stylistic and syntactic features.

3.3. Method for Web Propaganda Patterns Detection

After forming datasets and training a set of individual binary neural network models NN , detection of web propaganda patterns occurs. Scheme of method of web propaganda patterns detection by transformer neural networks is shown in Figure 5.

Input data of the method detection of web propaganda patterns by transformer neural networks are fine-tuned models NN , web content for analysis and threshold t .

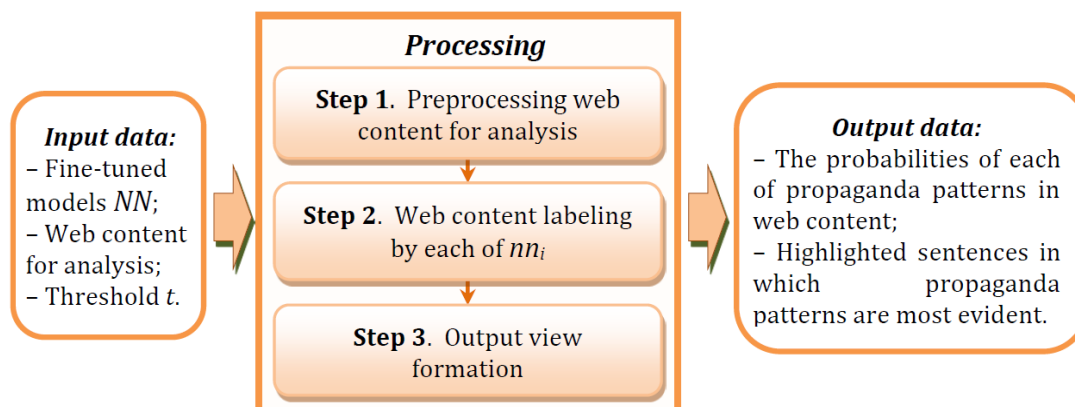


Figure 5: Scheme of method for web propaganda patterns detection

On Step 1, preprocessing of web content for analysis occurs, which includes of splitting into sentences, after which tokenization is performed [32, 33]. The result of web content splitting for analysis will be the representation (3):

$$S = \{s_1, s_2, \dots, s_n\}, \quad (3)$$

where s_j – j -th sentence in web content for analysis, n – count of sentence.

Step 2 performs web content labeling by each of nn_i . Each sentence s_j is evaluated separately by each of nn_i , and if the output value of the neural network model nn_i for sentence j exceeds the given threshold t – propaganda pattern p_i is considered to be manifested in sentence j . Accordingly, each sentence will be given a subset PP_j of the elements of the set P :

$$PP_j \subseteq P, PP_j = \{p_i | score_{i,j} > t\}, \quad (4)$$

where $score_{i,j}$ – the output value nn_i of the model for j -th sentence in $\{S\}$.

At Step 3, the formation of output view takes place, which is performed according to rules:

- if there are already manifestations of other propaganda patterns for sentence s_j , then such propaganda patterns are considered manifested in the text, however, the maximum value max_score_j will be displayed with highlighting:

$$max_i = \max_{p_i \in PP_j} score_{i,j}, \quad (5)$$

- if there are multiple sentences with the p_i propaganda pattern, the overall score of the manifestation in web content for analysis is calculated as the arithmetic mean:

$$Score = \frac{1}{\sum_{s_j \in SS_i} score_{i,j}}, SS_i = \{s_j | p_i \in PP_j\} \quad (6)$$

where SS_i – a set of sentences in which p_i is found.

Output data of the proposed method are probabilities of each of propaganda patterns in web content and highlighted sentences in which identified patterns are most evident [34].

The proposed in sections 3.1 – 3.2 methods are investigated experimentally in section 4.

4. Experiment

In accordance with purpose of research, problem of improving efficiency via dataset balancing arises, which can be mathematically represented as a problem of maximizing the F_1 metric:

$$m^* = \arg \max_m f(m), \quad (7)$$

where $f(m)$ – the value of the F_1 metric of the nn_i model obtained after fine-tuning on the dataset with the selected percentage value m .

The solution of the optimization problem will be carried out experimentally, changing the % of non-propaganda texts in the Training DataSet from 10% to 70% in steps of 20%.

For the experimental part, specialized software was created, consisting of 2 modules: a training module (without a graphical user interface) and a neural network validation module (the application is shown in Figure 6). The Python language, PyTorch libraries [35], transformers [36], datasets [37] were used to develop the training module. The PySide6 libraries [38], transformers, PyTorch were used to develop the validation module.

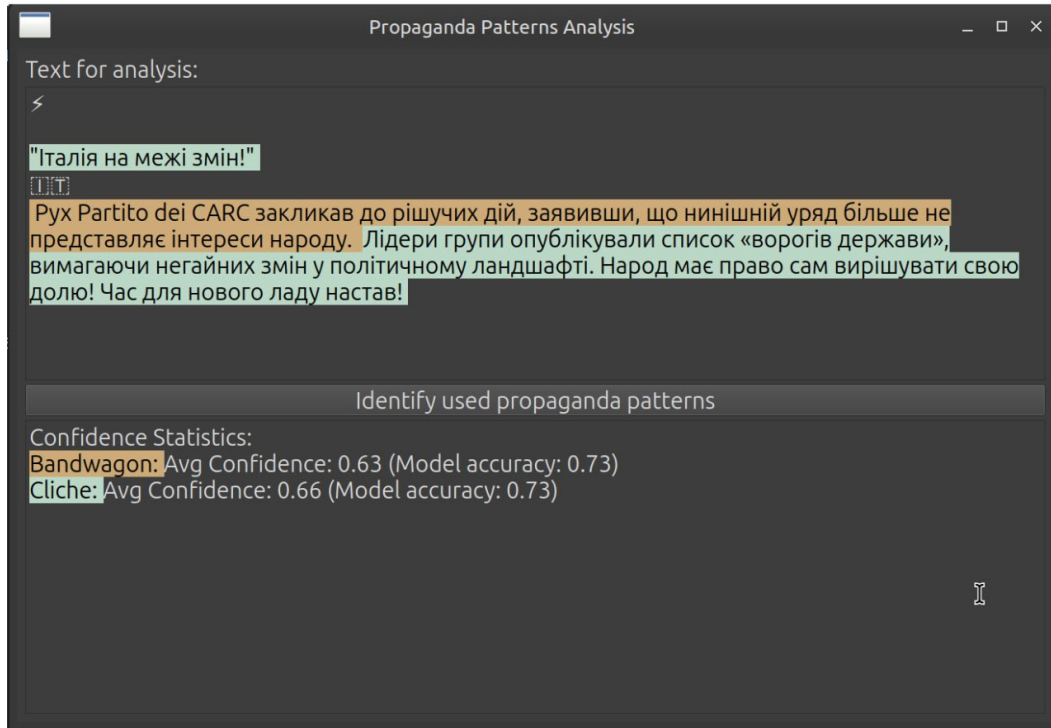


Figure 6: Web propaganda patterns detection by test software

Accordingly, the created test software obtained the results shown in Section 5.

5. Results

After filling the Training DataSet using the method described in section 3.1, data sets were obtained, the quantitative distributions of which are given in Table 2.

Table 2
Distribution of training dataset elements by percentage of texts without propaganda patterns (*m*)

Propa- ganda patterns	Target	Non target	Target	Non target	Target	Non target	Target	Non target
<i>m</i> %	10%		30%		50%		70%	
p_1	695	657	695	668	695	679	695	690
p_2	320	311	320	317	320	314	320	317
p_3	1032	954	1032	982	1032	1012	1032	1026
p_4	767	719	767	740	767	761	767	762
p_5	684	646	684	659	684	667	684	679
p_6	1131	1028	1131	1059	1131	1095	1131	1116
p_7	887	822	887	840	887	868	887	881
p_8	2541	2087	2541	2253	2541	2391	2541	2485
p_9	296	291	296	292	296	293	296	293
p_{10}	321	311	321	317	321	316	321	318

The Precision (P), Recall (R), F_1 metrics for fine-tuned individual binary neural network models at different percentage values of the parameter *m* on the test sample (20% of the Training DataSet, which did not participate in training) are given in Table 3. The Precision (P), Recall (R), F_1 metrics for fine-tuned individual binary neural network models at different percentage values of the parameter *m* on the training sample (80% of the Training DataSet, which participated in training) are given in Table 4.

Table 3

Results on test sample (20% of the training dataset that did not take part in training)

		10%			30%			50%			70%		
Propaganda patterns		P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
BERT	p_1	0.5105	0.5205	0.4979	0.6169	0.6173	0.6094	0.7365	0.7224	0.717	0.816	0.8002	0.7976
	p_2	0.7475	0.7456	0.746	0.7165	0.7139	0.7125	0.7668	0.7635	0.7618	0.8346	0.8333	0.8334
	p_3	0.6401	0.6391	0.639	0.6804	0.6803	0.68	0.7125	0.7122	0.7122	0.8096	0.7949	0.7937
	p_4	0.6162	0.6162	0.6162	0.6487	0.6484	0.6485	0.7165	0.7164	0.7161	0.8284	0.8159	0.8154
	p_5	0.6342	0.6343	0.6337	0.6969	0.6963	0.6947	0.7402	0.7332	0.7321	0.8431	0.8378	0.8374
	p_6	0.6655	0.6535	0.6555	0.6408	0.6406	0.6405	0.7718	0.7717	0.7713	0.8400	0.8359	0.8354
	p_7	0.5446	0.5436	0.5439	0.6609	0.649	0.6431	0.6941	0.6830	0.6804	0.7886	0.7810	0.7803
	p_8	0.5867	0.5591	0.5615	0.6878	0.6875	0.6875	0.7232	0.7109	0.7069	0.812	0.8047	0.806
	p_9	0.5839	0.5829	0.5686	0.6575	0.6501	0.6469	0.7314	0.7139	0.708	0.8305	0.8277	0.8269
	p_{10}	0.6241	0.6271	0.6227	0.6010	0.6017	0.6013	0.685	0.6864	0.6845	0.7819	0.7797	0.7771
Roberta	p_1	0.5184	0.527	0.5057	0.6056	0.6069	0.5993	0.7096	0.7052	0.7028	0.8067	0.7952	0.7932
	p_2	0.7403	0.7368	0.7373	0.7372	0.737	0.7368	0.7543	0.7521	0.7522	0.8578	0.8559	0.8560
	p_3	0.6506	0.6504	0.6504	0.6919	0.6914	0.6914	0.7096	0.7085	0.7083	0.8303	0.8168	0.8160
	p_4	0.5888	0.583	0.583	0.6571	0.652	0.652	0.6981	0.6982	0.6981	0.8014	0.8014	0.8014
	p_5	0.6527	0.6528	0.6527	0.6952	0.6941	0.6943	0.7455	0.7422	0.7418	0.8289	0.8289	0.8289
	p_6	0.6038	0.5984	0.6002	0.6564	0.6563	0.6562	0.732	0.7244	0.7203	0.7972	0.7969	0.7968
	p_7	0.5062	0.5067	0.5064	0.6027	0.6026	0.6025	0.6384	0.6373	0.6373	0.7489	0.7484	0.7484
	p_8	0.6071	0.5748	0.5764	0.7344	0.7344	0.7344	0.7591	0.7500	0.7478	0.7765	0.7500	0.7515
	p_9	0.5819	0.5829	0.5744	0.6035	0.603	0.6029	0.7323	0.7262	0.7241	0.8009	0.8010	0.8009
	p_{10}	0.5889	0.5847	0.5858	0.634	0.6356	0.6344	0.7284	0.7288	0.7267	0.7878	0.7881	0.7874

Table 4

Results of fine-tuned individual binary neural network models on the training set

		10%			30%			50%			70%		
Propaganda patterns		P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
BERT	p_1	0.7419	0.7188	0.7027	0.7854	0.7671	0.7608	0.8228	0.7954	0.7897	0.8986	0.8841	0.8829
	p_2	0.872	0.872	0.8719	0.9127	0.9095	0.9092	0.9228	0.9209	0.9208	0.9461	0.9441	0.9441
	p_3	0.8712	0.8712	0.8712	0.8577	0.8575	0.8576	0.9184	0.9167	0.9165	0.9266	0.922	0.9217
	p_4	0.8957	0.8945	0.8944	0.894	0.8927	0.8925	0.914	0.9099	0.9097	0.9059	0.8935	0.8925
	p_5	0.8694	0.8691	0.8689	0.8763	0.8704	0.8696	0.9103	0.9034	0.9028	0.9318	0.9254	0.9251
	p_6	0.8544	0.8532	0.8533	0.9048	0.9037	0.9037	0.8859	0.8797	0.879	0.9161	0.9096	0.9092
	p_7	0.7846	0.7837	0.7832	0.7877	0.7676	0.7624	0.854	0.8363	0.834	0.9164	0.9101	0.9096
	p_8	0.8694	0.8693	0.8693	0.9235	0.9235	0.9235	0.9455	0.943	0.9429	0.9234	0.9198	0.9197
	p_9	0.7297	0.7084	0.6988	0.7742	0.766	0.7631	0.852	0.8306	0.8277	0.8968	0.8809	0.8798
	p_{10}	0.6622	0.6503	0.6452	0.9114	0.9106	0.9106	0.8301	0.8301	0.8301	0.8886	0.8811	0.8807
Roberta	p_1	0.752	0.7358	0.7246	0.8103	0.7867	0.7801	0.8455	0.8289	0.826	0.9076	0.896	0.8951
	p_2	0.8851	0.8844	0.8843	0.9156	0.9146	0.9144	0.9532	0.953	0.953	0.9568	0.9562	0.9561
	p_3	0.908	0.9079	0.9079	0.9424	0.9413	0.9413	0.9409	0.938	0.9378	0.9484	0.945	0.9448
	p_4	0.9123	0.9121	0.9121	0.9358	0.9358	0.9358	0.9418	0.9399	0.9399	0.9572	0.9549	0.9548
	p_5	0.8948	0.8935	0.8932	0.9006	0.8995	0.8994	0.9277	0.923	0.9227	0.9547	0.9533	0.9532
	p_6	0.8853	0.8849	0.8847	0.9039	0.9037	0.9037	0.9546	0.9546	0.9546	0.9591	0.9568	0.9567
	p_7	0.8232	0.8207	0.82	0.8563	0.8556	0.8554	0.8878	0.883	0.8825	0.9299	0.9272	0.9271
	p_8	0.9287	0.9287	0.9287	0.9414	0.9412	0.9412	0.9485	0.947	0.9469	0.9591	0.9589	0.9589
	p_9	0.8409	0.8312	0.8294	0.8736	0.8721	0.8718	0.8895	0.8789	0.8779	0.92	0.9149	0.9147
	p_{10}	0.8863	0.8806	0.8803	0.9084	0.9064	0.9063	0.9286	0.9278	0.9278	0.9503	0.949	0.949

The Precision (P), Recall (R), and F_1 metrics for fine-tuned individual binary neural network models at different percentage values of parameter m on validation dataset are given in Table 5.

Table 5

Results of fine-tuned individual binary neural network models on the validation dataset

		10%			30%			50%			70%		
Propagand a patterns		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
B E R T	p_1	0.6686	0.6529	0.6283	0.6623	0.6457	0.6184	0.6689	0.6279	0.577	0.7046	0.6247	0.5551
	p_2	0.7828	0.778	0.7764	0.7901	0.7641	0.7571	0.7744	0.7362	0.7244	0.7802	0.7339	0.7199
	p_3	0.7582	0.7551	0.7536	0.7403	0.7362	0.7342	0.7506	0.7189	0.7072	0.7209	0.6292	0.579
	p_4	0.737	0.7202	0.7133	0.7412	0.7239	0.7171	0.7095	0.658	0.6309	0.7094	0.6025	0.5355
	p_5	0.7462	0.7384	0.734	0.7533	0.7296	0.7191	0.7536	0.7043	0.6829	0.7434	0.6525	0.6054
	p_6	0.7135	0.7046	0.7009	0.7485	0.7283	0.7219	0.7178	0.6714	0.6514	0.7167	0.6209	0.5709
	p_7	0.6523	0.6473	0.6415	0.6646	0.6291	0.6009	0.671	0.615	0.5715	0.6934	0.6109	0.5546
	p_8	0.7204	0.7172	0.7158	0.7518	0.7314	0.725	0.7575	0.6888	0.6647	0.703	0.6414	0.6091
	p_9	0.6531	0.6286	0.6074	0.6691	0.6485	0.6333	0.6941	0.6246	0.5802	0.704	0.6071	0.5451
	p_{10}	0.6284	0.6207	0.6122	0.7282	0.7069	0.6983	0.684	0.6724	0.6654	0.6872	0.619	0.5755
R O B E R T a	p_1	0.6771	0.6639	0.644	0.6843	0.6607	0.6332	0.6872	0.6544	0.6197	0.7123	0.6366	0.5755
	p_2	0.8057	0.8013	0.8	0.8001	0.7844	0.7805	0.7965	0.7757	0.7704	0.7897	0.7472	0.7354
	p_3	0.7738	0.7649	0.7619	0.7861	0.7611	0.7542	0.7569	0.7091	0.6917	0.764	0.6692	0.6313
	p_4	0.755	0.7506	0.7487	0.7859	0.7779	0.7756	0.752	0.7121	0.6975	0.7663	0.7032	0.6811
	p_5	0.781	0.7721	0.7684	0.7587	0.745	0.7387	0.7793	0.7394	0.7253	0.7624	0.7048	0.6811
	p_6	0.741	0.7267	0.722	0.7632	0.752	0.7489	0.7648	0.7441	0.7384	0.7467	0.6635	0.6307
	p_7	0.6669	0.6568	0.648	0.6732	0.6588	0.6477	0.6651	0.6278	0.5982	0.6895	0.6278	0.5867
	p_8	0.7539	0.7488	0.7472	0.7851	0.7694	0.7656	0.7796	0.7235	0.7076	0.749	0.7172	0.7068
	p_9	0.7156	0.688	0.6745	0.7092	0.6905	0.6807	0.7281	0.6695	0.6421	0.7121	0.644	0.6069
	p_{10}	0.7261	0.7069	0.699	0.7516	0.7293	0.7219	0.7386	0.6948	0.6778	0.736	0.6672	0.6378

Comparisons by the Accuracy metric (average value) for fine-tuned individual binary neural network models of the BERT and RoBERTa architectures at different percentage values of the parameter m on the Validation DataSet are shown in Figure 7.

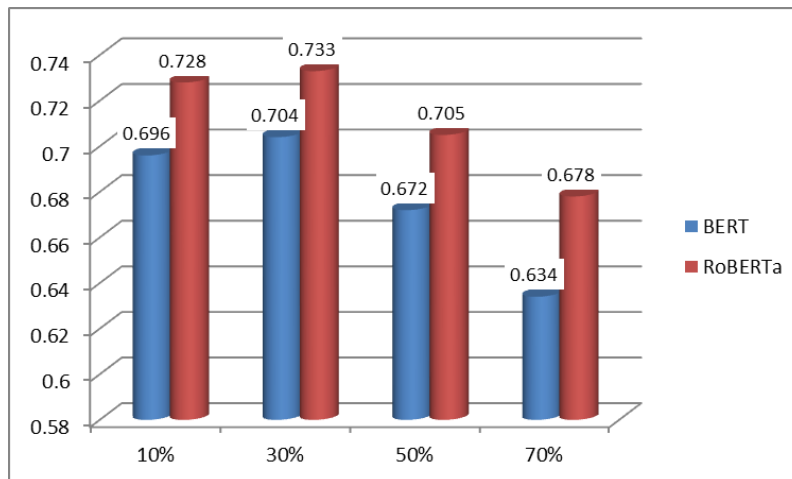


Figure 7: Comparison of fine-tuned models on validation dataset by Accuracy metric

Comparisons by F_1 metric (average value) for fine-tuned individual binary neural network models of the BERT and RoBERTa architectures at different percentage values of the parameter m on the Validation DataSet are shown in Figure 8.

It is also worth providing a table comparing the obtained results with the data of existing studies (Table 6).

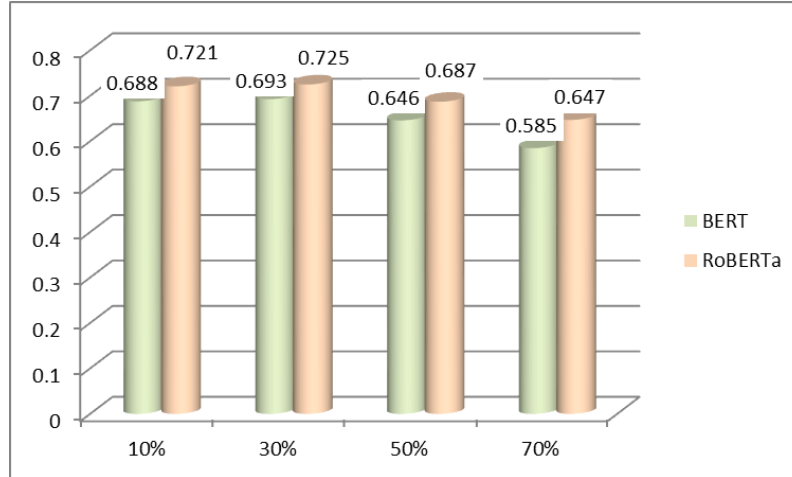


Figure 8: Comparison of fine-tuned models on validation dataset by F_1 metric

Table 6
Comparison of obtained results with existing research

Way of obtaining the result	Language	F_1
BERT, $m=10\%$	Ukrainian	0.688
BERT, $m=30\%$	Ukrainian	0.693
BERT, $m=50\%$	Ukrainian	0.646
BERT, $m=70\%$	Ukrainian	0.585
RoBERTa, $m=10\%$	Ukrainian	0.721
RoBERTa, $m=30\%$	Ukrainian	0.725
RoBERTa, $m=50\%$	Ukrainian	0.687
RoBERTa, $m=70\%$	Ukrainian	0.647
CRF+ BiLSTM [17]	multilingual	0.61
MultiProp-Baseline En-B [15]	Polish	0.625
RoBERTa [14]	English	0.602
Ensemble model [21]	English	0.604
Base + EDA [13]	not indicated	0.576

Therefore, the problem of improving efficiency via dataset balancing, given in the form (7), has a solution $m^* = 30$. An analysis of the obtained results is given in Section 6.

6. Discussion

In the presented results of testing models (Table 3) for detecting manipulative propaganda patterns on the test sample (20% of the training sample) with different percentages of texts without manipulations m (10%, 30%, 50%, 70%), one can observe a clear trend towards improving the performance of models with an increase in the value of the parameter m , i.e. with an increase in the percentage of texts without propaganda patterns in the training sample. For fine-tuned models based on BERT, it is seen that Precision, Recall and F_1 -measure for each category of propaganda patterns gradually increase from $m=10\%$ to $m=70\%$. For example, for the “Loaded Language” category, the F_1 -measure increases from 0.498 at $m = 10\%$ to about 0.798 at $m = 70\%$, which indicates a significant improvement in the model’s ability to distinguish target and non-target examples with an increase in the proportion of text samples without manipulations.

Comparing the performance of models for different propaganda patterns shows that some categories, such as “Glittering Generalities” and “Bandwagon”, “Cherry Picking”, have consistently high F_1 -measures as m increases, indicating that the characteristic features of these patterns are easier to separate with balanced training. In contrast, other categories, such as “Straw Man” and

“Thought-Terminating Cliche”, show relatively lower performance, which may be due to greater variability or subtlety of the linguistic features characterizing these patterns.

Similar analysis for RoBERTa-based models shows similar trends, with the overall performance being slightly higher compared to BERT models. This is explained by the more robust pre-training and optimized architecture of RoBERTa, which allows the model to generalize information better. The improvement in the evaluation indicators with an increase in the proportion of unmanipulated text samples highlights the importance of balancing the dataset to overcome the problem of class imbalance, which, in turn, contributes to more reliable and stable detection of propaganda patterns by individual binary neural networks. For the BERT neural network, on average, for the F_1 metric, the delta between $m = 10\%$ and $m = 30\%$ is $+0.048$, between $m = 30\%$ and $m = 50\%$, the delta is 0.063 , and between $m = 50\%$ and $m = 70\%$, the delta is 0.091 . At the same time, for the RoBERTa neural network, delta of $+0.0632$ is observed between $m = 10\%$ and $m = 30\%$, a delta of 0.056 is observed between $m = 30\%$ and $m = 50\%$, and delta of 0.082 is observed between $m = 50\%$ and $m = 70\%$.

The analysis of the data from Table 4 indicates the ability of neural network models to remember, and here, naturally, as in Table 3, there is a tendency for metrics to increase with increasing parameter m . For the BERT neural network, on average, for the F_1 metric, there is a delta between $m = 10\%$ and $m = 30\%$ of $+0.049$, between $m = 30\%$ and $m = 50\%$, there is delta of 0.02 , and between $m = 50\%$ and $m = 70\%$, there is delta of 0.031 . At the same time, for the RoBERTa neural network, a delta of $+0.028$ is observed between $m = 10\%$ and $m = 30\%$, delta of 0.022 is observed between $m = 30\%$ and $m = 50\%$, and delta of 0.024 is observed between $m = 50\%$ and $m = 70\%$. Accordingly, RoBERTa demonstrates a gradual increase in metrics, which indicates stable generalization due to the optimized architecture. BERT demonstrates somewhat jumpy increases, which may be due to the lower flexibility of its architecture in adapting to changes in the proportion of text samples without propaganda patterns.

For the RoBERTa neural network, when detecting manipulation patterns “Glittering Generalities”, “Appeal to Fear”, “FUD”, “Bandwagon”, “Whataboutism”, an F_1 value of more than 0.95 is observed. For the BERT neural network, a value above 0.95 is observed only for “FUD”. In general, the use of different values of the parameter m affects the ability of neural networks to remember the features of the training set. However, the metrics calculated on the training data allow us to assess how well the model remembered this data, but do not give a complete picture of its ability to generalize new information.

The most relevant estimates of the experiment are given in Table 5, since here the model was validated on data that did not participate in training, and which contain equally represented propaganda patterns and texts without such patterns.

According to Table 5 and Figures 7 and 8, at the parameter $m=30\%$ the metrics demonstrate the highest result, where the average value of the Accuracy metric is 0.733 for the RoBERTa neural network, and 0.704 for the BERT architecture. The F_1 metric for RoBERTa is 0.725 , and for the BERT architecture – 0.693 . This suggests that the initial addition of data allows to increase the metrics, but then the effect is smoothed out or even worsened due to overloading with less useful information, such as texts without propaganda patterns. Accordingly, while neural networks show a tendency to better distinguish propaganda patterns at higher values of m during training, testing on a balanced validation set refutes the hypothesis that the higher the resolution of the training data, the better the generalization ability of the neural network model. It is possible that as the proportion of m increases, the models are overtrained due to the lack of unique values inherent in each of the web propaganda patterns. The conclusion that the $m^*=30$ found is also confirmed by the minimum mean deviation between the test data for $m=30$ (Table 3 and Table 5) for both the BERT architecture neural network (0.05) and RoBERTa (0.06).

The comparison with analogues is carried out in Table 6, and for the purity of the comparison of the developed approach and existing analogues, the F_1 value was taken specifically on the validation data. Accordingly, the highest F_1 indicator for the RoBERTa architecture at $m=30\%$ is 0.725 , which is 0.1 higher than the analogue described in [15]. Therefore, the task of improving the efficiency of detecting propaganda patterns in web texts using transformative neural networks through optimizing the balancing of the dataset has been fully implemented and experimentally proven.

However, the proposed approach has limitations. In this study, an approach at the sentence level was used. This may have an impact on the quality of detecting propaganda patterns, which may work at the level of paragraphs or even entire texts, rather than individual sentences. Also, a single sentence may be neutral in itself, but in the context of propaganda text its meaning changes. These issues will be addressed in further research. There are also limitations at the level of the data source.

The manual labeling used in the dataset may contain subjective judgments, which affects the training of the model.

Conclusions

In the paper, a proposed approach for improving efficiency of web propaganda patterns detection by transformer neural networks is presented. Approach consists of sequential use of three developed methods: method for dataset balancing, method for fine-tuning individual binary neural network models and method for detecting web propaganda patterns. Compared to existing analogues, the use of proposed approach allowed achieving an efficiency increase of 0.1 by F_1 metric when detecting propaganda patterns in web texts using transformer neural networks due to dataset balancing optimization.

In addition to the training dataset, consisting of texts with target propaganda pattern in the target category, as well as texts without any propaganda patterns and texts with other propaganda patterns, without target, a validation dataset was built, which consists equally of all types of web propaganda patterns and texts without propaganda. This allows us to determine whether the model does not confuse patterns with each other and is able to detect them independently of each other, which is critically important for the patterns detection task.

An analysis of the impact of the balance of the training sample on the effectiveness of propaganda pattern detection models in social media was performed, which showed that of the considered options for forming the training dataset with different percentages of texts without manipulations (10%, 30%, 50% and 70%), the highest results were achieved at 30% using the RoBERTa neural network, and are 0.725 according to the F_1 metric. The results obtained contribute to a deeper understanding of the role of training sample balancing in improving propaganda pattern detection algorithms and can be used to increase the reliability of automated information space analysis systems.

Building a validation dataset that contains an equal number of texts with all types of propaganda patterns, as well as neutral texts, provides a fair assessment of the performance of the models. This prevents bias towards the most represented classes and allows for more accurate performance metrics for each individual pattern. In addition, this approach allows for the identification of potential relationships between different types of manipulation, since texts can contain multiple patterns at the same time.

Analyzing the impact of parameter that determines proportion of texts without web propaganda patterns allows assessing how the models ability to distinguish propaganda patterns from neutral texts and texts with other propaganda patterns. This allows finding the optimal ratio of dataset classes to increase the overall effectiveness for detecting web propaganda patterns.

The proposed approach has the limitation of analyzing at the sentence level, which may not take into account the broader context of propaganda patterns at the paragraph or whole text level. In addition, the use of manual data labeling may contain subjective judgments, which affects the training of the model.

Acknowledgements

This study was conducted using dataset made available by UNLP 2025 Shared Task initiative (GitHub repository) [24]. Authors are grateful to organizers and contributors for compiling and sharing this resource, which supports ongoing research in propaganda detection techniques.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] B. M. Almotairy, M. Abdullah, D. H. Alahmadi, Dataset for Detecting and Characterizing Arab Computation Propaganda on X, Data in Brief. (2024) 110089. doi:10.1016/j.dib.2024.110089.
- [2] A. Horák, R. Sabol, O. Herman, V. Baisa, Recognition of propaganda techniques in newspaper texts: Fusion of content and style analysis, Expert Systems with Applications. (2024) 124085. doi:10.1016/j.eswa.2024.124085.
- [3] M. Moore, T. Colley, Two International Propaganda Models: Comparing RT and CGTN's 2020 US Election Coverage, Journalism Practice. (2022) 1–23. doi:10.1080/17512786.2022.2086157.
- [4] M. Bösch, T. Divon, The sound of disinformation: TikTok, computational propaganda, and the invasion of Ukraine, New Media & Society. 26.9 (2024) 5081–5106. doi:10.1177/14614448241251804.
- [5] I. Krak, M. Molchanova, V. Didur, O. Sobko, O. Mazurets, O. Barmak, Method of semantic features estimation for political propaganda techniques detection using transformer neural networks, CEUR Workshop Proceedings, 3917 (2025) 286–297. URL: <https://ceur-ws.org/Vol-3917/paper56.pdf>
- [6] M. Karas, T. I. Profant, Propaganda Model in Slovak Media Space: A Case Study, Hum. Aff. 35(1). (2024) 36–60. doi:10.1515/humaff-2024-0062.
- [7] F. Alam, M. R. Biswas, U. Shah, W. Zaghouani, G. Mikros, Propaganda to Hate: A Multimodal Analysis of Arabic Memes with Multi-agent LLMs, In: Lecture Notes in Computer Science, Springer Nature Singapore, 2024, p. 380–390. doi:10.1007/978-981-96-0576-7_28.
- [8] G. Yumnam, Y. Gyanendra, C. I. Singh, A systematic bibliometric review of the global research dynamics of United Nations Sustainable Development Goals 2030, Sustain. Futures 7 (2024) 100192. doi:10.1016/j.sftr.2024.100192.
- [9] C. Işık, S. Ongan, D. Ozdemir, J. Yan, O. Demir, The sustainable development goals: Theory and a holistic evidence from the USA, Gondwana Research. 132 (2024) 259–274. doi:10.1016/j.gr.2024.04.014.
- [10] S. Sorooshian, The sustainable development goals of the United Nations: A comparative midterm research review, Journal of Cleaner Production. 453 (2024) 142272. doi:10.1016/j.jclepro.2024.142272.
- [11] R. Raman, V. Kumar Nair, P. Nedungadi, A. Kumar Sahu, R. Kowalski, S. Ramanathan, K. Achuthan, Fake news research trends, linkages to generative artificial intelligence and sustainable development goals, Heliyon (2024) e24727. doi:10.1016/j.heliyon.2024.e24727.
- [12] G. Faye, B. Icard, M. Casanova, J. Chanson, F. Maine, F. Bancelhon, P. Égré, Exposing propaganda: an analysis of stylistic cues comparing human annotations and machine classification, arXiv preprint arXiv:2402.03780 (2024). URL: <https://arxiv.org/abs/2402.03780>.
- [13] W. Li, S. Li, C. Liu, L. Lu, Z. Shi, S. Wen, Span identification and technique classification of propaganda in news articles, Complex & Intelligent Systems. (2021). doi:10.1007/s40747-021-00393-y.
- [14] M. Abdullah, O. Altit, R. Obiedat, Detecting Propaganda Techniques in English News Articles using Pre-trained Transformers, 13th International Conference on Information and Communication Systems (2022) 301–308. doi:10.1109/ICICS55353.2022.9811117.
- [15] F. Aldabbas, S. Ashraf, R. Sifa, L. Flek, MultiProp Framework: Ensemble Models for Enhanced Cross-Lingual Propaganda Detection in Social Media and News using Data Augmentation, Text Segmentation, and Meta-Learning, Proc. 1st Workshop NLP Lang. Using Arab. Script (2025) 7–22. URL: <https://aclanthology.org/2025.abjadnlp-1.2>.
- [16] J. Szwoch, M. Staszko, R. Rzepka, K. Araki, Limitations of Large Language Models in Propaganda Detection Task, Applied Sciences. 14.10 (2024) 4330. doi:10.3390/app14104330.
- [17] P. N. Ahmad, L. Yuanchao, K. Aurangzeb, M. S. Anwar, Q. M. u. Haq, Semantic web-based propaganda text detection from social media using meta-learning, Service Oriented Computing and Applications. (2024). doi:10.1007/s11761-024-00422-x.
- [18] D. G. Jones, Detecting Propaganda in News Articles Using Large Language Models, Engineering 2.1 (2024) 01–12. doi:10.33140/eoa.01.02.10.
- [19] A. A. Mustafa, C.-Y. Lin, M. Kakinaka, Detecting market pattern changes: A machine learning approach, Finance Research Letters. (2021) 102621. doi:10.1016/j.frl.2021.102621.
- [20] A. M. U. D. Khanday, Q. R. Khan, S. T. Rabani, M. A. Wani, M. ELAffendi, Propaganda Identification on Twitter Platform During COVID-19 Pandemic Using LSTM, In: Advances in

- Cybersecurity, Cybercrimes, and Smart Emerging Technologies, Springer International Publishing, Cham (2023) 303–314. doi:10.1007/978-3-031-21101-0_24.
- [21] M. Abdullah, D. Abujaber, A. Al-Qarqaz, R. Abbott, M. Hadzikadic, Combating propaganda texts using transfer learning, *Int J Artif Intell* ISSN. (IJ-AI) 12.2 (2023) 956. doi:10.11591/ijai.v12.i2.pp956-965.
 - [22] F. J. Rodrigo-Ginés, J. Carrillo-de-Albornoz, L. Plaza, Hierarchical Modeling for Propaganda Detection: Leveraging Media Bias and Propaganda Detection Datasets, *IberLEF@ SEPLN* (2023). URL: <https://ceur-ws.org/Vol-3496/dipromats-paper7.pdf>.
 - [23] UNLP Workshop, UNLP-2025 Shared Task Data, 2025. URL: <https://github.com/unlp-workshop/unlp-2025-shared-task/tree/main/data>.
 - [24] UNLP Workshop, UNLP-2025 Shared Task, 2025. URL: <https://github.com/unlp-workshop/unlp-2025-shared-task>.
 - [25] O. Sobko, O. Mazurets, M. Molchanova, I. Krak, O. Barmak, Method for analysis and formation of representative text datasets, *CEUR Workshop Proceedings*, 3899 (2024) 84-98. URL: <https://ceur-ws.org/Vol-3899/paper9.pdf>
 - [26] I. Krak, O. Zalutska, M. Molchanova, O. Mazurets, E. Manziuk and O. Barmak, Method for neural network detecting propaganda techniques by markers with visual analytic, *CEUR Workshop Proceedings*, 3790 (2024) 158-170. URL: <https://ceur-ws.org/Vol-3790/paper14.pdf>.
 - [27] Hugging Face, google-bert/bert-base-multilingual-cased, 2025. URL: <https://huggingface.co/google-bert/bert-base-multilingual-cased>.
 - [28] Hugging Face, youscan/ukr-roberta-base, 2025. URL: <https://huggingface.co/youscan/ukr-roberta-base>.
 - [29] Hugging Face, Models and Libraries Documentation, 2025. URL: <https://huggingface.co/docs/hub/models-libraries>.
 - [30] N. M. Gardazi, A. Daud, M. K. Malik, BERT applications in natural language processing: a review, *Artif. Intell. Rev.* 58 (2025) 1–49. doi:10.1007/s10462-025-11162-5.
 - [31] J. Carreras Timoneda, S. Vallejo Vera, BERT, RoBERTa or DeBERTa? Comparing Performance Across Transformer Models in Political Science Text, *The Journal of Politics* (2024). doi:10.1086/730737.
 - [32] Y. Krak, O. Barmak, O. Mazurets, The Practice Investigation of the Information Technology Efficiency for Automated Definition of Terms in the Semantic Content of Educational Materials, *CEUR Workshop Proceedings*, 163 (2016) 237–245. URL: doi:10.15407/pp2016.02-03.237.
 - [33] O. Barmak, O. Mazurets, I. Krak, A. Kulias, A. Smolarz, L. Azarova, K. Gromaszek, S. Smailova, Information technology for creation of semantic structure of educational materials, *Proceedings of SPIE – The International Society for Optical Engineering*, 11176 (2019) 1117623. doi:10.1117/12.2537064
 - [34] O. Kovalchuk, V. Slobodzian, O. Sobko, M. Molchanova, O. Mazurets, O. Barmak, I. Krak, N. Savina, Visual Analytics-Based Method for Sentiment Analysis of COVID-19 Ukrainian Tweets, *Lecture Notes on Data Engineering and Communications Technologies*, 149 (2023) 591–607. doi:10.1007/978-3-031-16203-9_33.
 - [35] PyTorch, PyTorch Official Website, 2025. URL: <https://pytorch.org/>.
 - [36] I. Krak, V. Didur, M. Molchanova, O. Mazurets, O. Sobko, O. Zalutska and O. Barmak, Method for political propaganda detection in internet content using recurrent neural network models ensemble, *CEUR Workshop Proceedings*, 3806 (2024) 312-324. URL: https://ceur-ws.org/Vol-3806/S_36_Krak.pdf.
 - [37] PyPI, Datasets, 2025. URL: <https://pypi.org/project/datasets/>.
 - [38] PyPI, PySide6, 2025. URL: <https://pypi.org/project/PySide6/>.